AGILIZING INTERACTION FLOW MODEL LANGUAGE BY USING KANBAN FOR COVERING WEB APPLICATION SHORT LIFECYCLE

SAFAA JAWAD ALWAN AL-JABERI

A dissertationsubmitted in fulfilmentof the
requirements for the award of the degree of
Master of Computer Science

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

JUNE 2019

ACKNOWLEDGEMENT

First of all, I would like to warmly express my utmost gratitude to Allah S.W.T for His blessing and given strength during the completion of this research.

I would also like to sincerely thank my supervisor **Assoc. Prof. Dr. Dayang Norhayati Binti Abang Jawawi**, for her continuous motivation, advices, encouragement and support from the beginning to the end of my studies, till I was able to develop a deep understanding of the research topic.

Finally my special thanks to my family and my friends for their love, care for their support and cheering me up at those difficult time.

ABSTRACT

The World Wide Web has turned into a significant stage for the conveyance of an assortment of complex corporative applications in numerous spaces. Other than its disseminated viewpoint, these Web applications require consistent enhancements in the lifecycle, convenience, execution, security, and versatility. In any case, by far most of the previously mentioned applications are as yet being produced with an ad hoc approach, contributing for lifecycle issue, maintenance, quality, and reliability. Web engineering methods become to main directions to develop web applications systematically, focusing on methods, processes, techniques, and tools applied in different abstraction levels, from the conception to development, evaluation, and maintainability. Most web engineering methods have lack to support the process development lifecycle including IFML as the latest method in one hand. Meanwhile, an agile approach is used to support a lifecycle for web applications by integrating an agile approach and web engineering method for designing a web mining system. The goal of this research is to evaluate an IFML to cover short lifecycle for developing complex web applications. There are three steps in this development process. In the first step, an investigation of IFML lifecycle through design web mining system. In the second step, an integrated IFML with Kanban for short lifecycle to propose KIFML. In the last step, KIFML is evaluated using the Technology Acceptance Model (TAM) to identify the acceptance of respondents towards the methods. Finally, a comparison between an integrated IFML and present IFML by using case study. Then the result showed capabilities of evaluated IFML for developing web applications short lifecycle.

ABSTRAK

Jaringan Sejagat telah menjadi peringkat yang penting untuk penyampaian pelbagai kesatuan aplikasi kompleks di dalam banyak bidang. Selain daripada pandangan yang meluas, aplikasi sesawang ini memerlukan peningkatan yang konsisten dalam kitaran hayat, kemudahan, pelaksanaan, keselamatan, dan serba boleh. Walau bagaimanapun, setakat ini kebanyakan aplikasi yang telah disebutkan sebelum ini dihasilkan dengan pendekatan yang diperlukan, menyumbangkan kepada isu kitaran hidup, penyelenggaraan, kualiti dan kebolehpercayaan. Kaedah kejuruteraan sesawang menjadi penunjuk utama untuk membangunkan aplikasi sesawang yang sistematik, fokus kepada kaedah, proses, teknik, dan alat yang digunakan pada peringkat abstrak yang berbeza, dari konsep kepada pembangunan, penilaian dan pemeliharaan. Kebanyakan kaedah kejuruteraan sesawang kurang menyokong proses pembangunan kitaran hayat termasuk IFML sebagai kaedah terkini dalam satu tangan. Sementara itu, metodologi agil digunakan untuk menyokong kitaran hayat aplikasi sesawang dengan menyepadukan kaedah agil dan kaedah kejuruteraan sesawang untuk mereka bentuk sistem perlombongan sesawang. Matlamat penyelidikan ini ialah penilaian kepada IFML untuk menampung kitaran hayat pendek bagi pembangunan aplikasi sesawang yang kompleks. Terdapat tiga langkah dalam proses pembangunan ini. Dalam langkah pertama, siasatan kepada kitaran hayat IFML melalui sistem perlombongan sesawang reka bentuk. Dalam langkah kedua, kesatuan IFML bersama Kanban untuk menampung kitar hayat yang pendek. serta pengesahan dengan menggunakan TAM untuk mengenalpasti tahap penerimaan responden bagi kaedah terseput Dalam langkah terakhir, penilaian kepada IFML baru dalam kitaran hayat proses pembangunan. Akhirnya, perbandingan antara IFML yang dinilai dan IFML sedia ada dengan menggunakan kes kajian, dan kemudian hasil akan menunjukkan keupayaan IFML yang dinilai untuk membangunkan aplikasi sesawang kitaran hayat pendek.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AM	-	Agile Modeling
ASD	-	Agile Software Development
AWE	-	Agile Web Engineering
CIM	-	Computation Independent Model
CMMI	-	Capability Maturity Model Integration
DSDM	-	Dynamic Systems Development Method
FDD	-	Feature Driven Development
IFML	-	Interaction Flow Modeling Language
MDA	-	Model Driven Architecture
MDD	-	Model Driven Development
MDE	-	Model-Driven Engineering
MDWE	-	Model-driven Web engineering
MobML	-	Mobile Modeling Language
ODM	-	Ontology Definition Meta-Model
OMG	-	Object Management Group
OOHDM	-	Object Oriented Hypermedia Design Method
OpenAPI	-	Open Application programming interface
OWL	-	The Web Ontology Language
PIM	-	Platform Independent Model
PLE	-	Product Line Engineering
SLR	-	Systematic Literature Review
SoaML	-	Service Oriented Architecture Modeling Language
SPEM	-	Software Process Engineering Metamodel
SSE-CMM	-	System Security Engineering Capability Maturity Model
UML	-	Unified Modeling language

CHAPTER 1

INTRODUCTION

1.1 Introduction

According to the recognition of developers and directors in creating applications become functional by the new technologies, complexity of web applications. The need for engineering is identified and highlighted in web document engineering and website engineering (White, 1996), (Powell *et al.*, 1998). Good web development requires multidisciplinary efforts and does not fit properly in any of the existing disciplines which web engineering explicitly recognizes.

As defined, web engineering is the application of systemic, disciplined and quantifiable approaches to web - based applications development, operation and maintenance (Murugesan *et al.*, 2001). Several web engineers use methods, information engineering techniques, hypertext and hypermedia design, data recovery and data extraction, graphics and interaction design, etc. Software and web engineering involve the development of software and its lifecycle (including maintenance and evolution) (Kienle and Distante, 2014) that partially cover the hypermedia systems' lifecycle and are highly centered on the configuration of such systems (De Koch, 2001).

Through a combination of some methods with other web engineering methods or adding other methods to support the lifecycle and improve the process development. The prominence of and need for Web Engineering is now sensibly established, through a consensus among specialists on the essential differences in the characteristics of Web applications and conventional software (Murugesan *et al.*, 2001).

A good Web development must utilise published information disciplines for global access, publishing paradigms, and legal, social and ethical issues must be taken into account and not dominated by narrow points of view. This multidisciplinary nature of web applications is recognized as a web engineering response (Murugesan *et al.*, 2001).

Model-driven engineering is defined as a model-driven web engineering (MDWE) in the field of web application development. It is especially useful because web technologies and platforms are constantly evolving (Kraus *et al.*, 2007). The most essential commitments of MDWE approaches to deal with the more broad zone have been for the most part the recognizable proof of demonstrating concerns which are particular to the Web domain, for instance, navigation and user interface/interaction (Rossi *et al.*, 2016).

During a few years ago, several researchers aimed to make integration between web engineering models to optimize some web applications, by creating new models that can be approved. One of these models that adopted in 2013 is an Interaction Flow Modeling Language (IFML) that an Object Management Group (OMG) as the standard language to describe application interaction aspects and optimize user experience (Frajták *et al.*, 2015). The first version of IFML (IFML 1.0) has only (2015) been officially published (Group). It expresses the components that reside in the same interface window in the application and which of them are visible at the same time.

Throughout web applications development that embrace variations on the rational unified method, or structured analytic thinking and style approaches are increasingly appeared differently in relation to, and supplanted by, so-called agile processes, illustrated in terms of atiny low set of development necessities and practices, e.g., the utilization of little development groups, short development lifecycles, and specializing in the production of software deliverables as opposition documentation or models (Ge *et al.*, 2006).

Agile – devoting “the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion” as mentioned in ("Oxford English

Dictionary,") – software improvement strategies are endeavoring to offer once again by a response to the excited business network requesting lighter weight alongside quicker and nimbler software advancement processes.

The three characteristics enclosed of agile processes comprises acceptive the changes of needs can occur throughout development, progressive development of needs during the lifecycle and stress on individuals and their interactions instead of consistence to a specific set of process objectives.

In principle, the lifecycle is there anticipate messes up that will cost beyond a reasonable doubt later on in the project. one of the agile methods is Kanban that term came into existence using the flavours of a visual card, “signboard,” or “billboard” (Beck *et al.*, 2001), it acknowledges potential bottlenecks within the process and settle them thus work will flow through it cost-effectively at an optimal speed or turnout. Furthermore, it signals the system to indicate a workflow that limits Work In Progress (WIP). Work In Progress is the number of task items that a team is currently working on (Tanner and Dauane, 2017). It frames the capacity of team’s workflow at any moment. The core property of Kanban consists of Visualize Workflow, Limit Work in Progress (WIP) and the Lead Time measurement. This characteristic of Kanban is an immediate execution of a Lean Pull Scheduling System.

Therefore, the requirement to prevent tension on resources in the workflow state of applications, reduce waiting time and decrease dependence on achieving a task is an essential point for developing web engineering methods, including the integration of IFML with other web engineering methods or agile methods to support the process of the web development short lifecycle.

1.2 Problem Background

One of the most important commons of the applications problems are developed without following any development methodology and occasionally without following any project management methodology (Mnkandla and Dwolatzky, 2004).

The methodologies or development processes for internet application development that change one to ace the over all complexness of internet application design. The developer should have the capability to License figure and break down requirements, build associate interpretation of them update the application by selecting the appropriate innovations and instruments to relate application plans, test and validate the result, operate, maintain and evolve the application as required (Casteleyn *et al.*, 2009).

The difficulty of Model-Driven Web Engineering (MDWE) approaches to accumulate fast feedback from customers because of sharing a top-down approach, it starts by modeling application substance then it defines a steering schema, finally refines the latter to induce a presentation and wealthy conduct determinations (Rivero *et al.*, 2011). Moreover, each of web engineering methods is defined to particular determinations, therefore, many lacks in it, for instance, the open issue of boosting the whole lifecycle in process development (Wakil and Jawawi, 2017a). Moreover, most current approaches for MDWE provide simply a halfway use of the MDA (Model Driven Architecture) pattern.

Most current approaches for MDWE offer solely a partial application of the MDA (Model Driven Architecture) model. In addition, metamodels and transformations don't seem to be perpetually created categorical and metamodels are frequently overly broad or do not contain adequate knowledge for the automated code generation (Kraus *et al.*, 2007), although, analysis ways projected, the danger is web engineering methods cannot cover the whole web development lifecycle particularly analysis part and therefore the developers generally don't meet users' expectations (Wakil and Jawawi, 2018).

Several model-driven approaches practically disregard stakeholders' participation (Grigera *et al.*, 2012) because they represent a step forward to decrease both development time and work at a higher level of deliberation. In addition, there are new challenges apply to the models and deliberations for capturing new application requirements in addition to systems and technology (Brambilla and Ceri, 2018). furthermore, existing instruments around Open Application programming interface (OpenAPI) remaining difficult for non-developers to survey service features (Koren and Klamma, 2018).

Moreover, build application code from high-level characterizations require model-to-model and/or model-to-text transformations of Model Driven Development (MDD). Producing such transformations is in itself complicate job, it needs mastering meta-modeling, ad hoc transformation languages, and various development tools (Bernaschina, 2017).

Generating new interfaces that serve user behaviors, and meet its standardisation and simplification requirements become a major trend to integrate MDWE with other tools and methods, for instance, integrate models and owl ontologies to derive UIs web-Apps (Laaz and Mbarki, 2016b). As well as, integrate several recent methods such as IFML that adapted by OMG (Group) and has a good method best practice, composed of a UML profile and support rich interface, but cannot fully support the web development lifecycle (Wakil and Jawawi, 2017a) with other tools to achieve supporting web development lifecycle.

In some literature, prior studies that propose the agile methods support lifecycle phases process (Alpaslan and Kalıpsız, 2016) for web applications. The requirements to avoiding stress on resources at a workflow state of applications, reducing wait time and decreasing dependencies in completing a task (Kniberg and Skarin, 2010) due to these issues, develop an Agile Web Engineering (AWE) process to tackle the problems related with the development of Web-based applications (McDonald and Welland, 2001a). The Kanban method must meet the challenge of a new project combined with a modern technology and methodology for development without overloading the team. It was adapted to develop maturity levels, tools and

processes as criteria to point out once the team is ready to move to the subsequent level. (Hofmann *et al.*, 2018).

While fundamental Web investigation apparatuses are widespread and give measurements concerning website navigation, no methodologies exist for mixing such insights with knowledge concerning the Web application structure, content and semantics (Bernaschina *et al.*, 2017a) to keep the idea of user behavior, indicate the advantages of integrating Web application models with runtime navigation logs.

There is still a scarcity of explicit studies and analysis (Torrecilla-Salinas *et al.*, 2016) of an Agile approach to to satisfy varied Capability Maturity Model Integration maturity levels objectives might be in a position for a corporation developing Web systems. Furthermore, a vital limitation lacks upfront planning, sufficient documentation, and predictability make Agile software development more stressful (Agrawal *et al.*, 2016). On the other hand, other researchers used what is commonly referred to as mockups (user interface prototypes) as an approach to start the modeling process in an integrated agile MDWE method (Rivero, 2014).

The Web is an extensive repository of information that grows rapidly. The intensive growth of information poses many new challenges to web researchers, including high data dimensionality and highly volatile and constantly evolving content. (Sunil Kumar and Suvarchala, 2012). Web information appearance is an essential disadvantage in current information extraction trends. Traditional plans to achieve the massive measurements of web - based data basically assume a text-oriented, keyword - based view of web pages. To obtain the information that need high - potential web mining techniques to overcome the basic problems (Yu *et al.*, 2002).

However, the objective of web mining is to search for useful web data patterns by collecting and analyzing information to gain insight into trends, industry, and users, There is a complex issue while utilising web mining with engineering to process web applications (Nayak and Reddy, 2013) because of the assortment of dialects and advancements that are contemporary used to acknowledge them.

The needs of the process associated with web data products associated with Web Engineering framework. For example, web configuration management data, web application classification, web development methodologies, web development process. The knowledge of web mining techniques used in these services are in fact different types of web applications. Patterns extracted from applying web mining techniques on web data can be used to maintain websites by rising their necessities which is able to consequently increase the general profit of the business or the industry that the maintained website belongs to (Chung, 2006).

Thus, the needs to avoiding stress on resources at a workflow state of applications, reducing wait time and decreasing dependencies in completing a task, therefore, that is an important point by which to improve web engineering methods including IFML integrating with other web engineering methods or agile methods to support process of the web development lifecycle.

1.3 Problem Statement

Web engineering methodologies become main directions to develop web applications systematically. The main problem of this dissertation is web engineering methods including the latest method IFML have lack to cover the whole process development lifecycle (Wakil and Jawawi, 2017a). In addition, the needs to avoiding stress on resources at a workflow state of applications, reducing wait time and decreasing dependencies in completing a task.

The above premises lead to the main research question:

"How to represent the complex web application system development lifecycle at domain engineering and the modeling at web application to enable an effective development short lifecycle?"

The supporting research questions are:

i. How to represent the domain modelling of IFML web engineering method for web mining system?

ii. How to avoid the stress on resources at a workflow state of applications using agile approaches?

iii. How to reduce the waiting time and decreasing dependencies in completing a task?

1.4 Aim of Study

Agile methodology is one of important way to improve web engineering methods, and Kanban method in agile development shows a promising solution in industry. The aim is to study the applicability of agile methodology to improve web engineering methods to cover short lifecycle.

1.5 Research Objectives

The objectives of our research focusing on three vital points those are:

(a) To investigate IFML lifecycle through design web mining system.
(b) To integrate IFML with agile methodology to cover short lifecycle.
(c) To compare the present IFML with new IFML in the process development lifecycle.
(d) To evaluate the acceptance of new IFML method using the Technology Acceptance Model (TAM).

1.6 Research Scopes

The scope of research illustrated below:

i) Web page achieved that includes the result of using web mining system.

ii) This research focuses on using the IFMLEditor tool in the study.

1.7 Significant of Study

The most significant of research to evualate web engineering method by integrated with agile methods, design complex web application like web mining system by utilizing web engineering methods systematically and short time, in addition to, specify which of tools can achieve new web application techniques. Thus this research has a great involvement in development process of complex web applications, it helps to cover the development process short lifecycle. It also helps to alleviate types of challenges, communication, and collaboration in web engineering methods.

1.8 Research Organization

Chapter 2 discuss the web engineering methods and its challenges, IFML method and its lack, agile methods and its challenges, identified web mining as a domain.

Chapter 3 the research methodology is conducted in achieving the dissertation objectives and scopes. On case study is used web mining as domain, tools to develop present IFML lifecycle.

REFERENCES

Abd, S. (2016). Core Asset Variability Modeling Approach for Software Product Line Development. Universiti Teknologi Malaysia.

Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. (2017). Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.

Acerbis, R., Bongio, A., Brambilla, M., and Butti, S. (2015). Model-driven development based on omg's IFML with webratio web and mobile platform. Paper presented at the International Conference on Web Engineering, 605-608.

Agrawal, A., Atiq, M. A., and Maurya, L. (2016). A Current Study on the Limitations of Agile Methods in Industry Using Secure Google Forms. Procedia Computer Science, 78, 291-297.

Alpaslan, G., and Kalıpsız, O. (2016). Model driven web application development with agile practices. arXiv preprint arXiv:1610.03335.

atlassian. from https://www.atlassian.com/agile/kanban

Bancora, M., Brambilla, M., Ripamonti, D., and Vaccarella, A. (2015). Personal, Social and Event Organization Through Web and Mobile Apps: The Fluxedo Case. Paper presented at the International Conference on Web Engineering, 609-612.

Barcelona, M., García-Borgoñón, L., López-Nicolás, G., Ramos, I., and Escalona, M. J. (2017). Applying a model-based methodology to develop web-based systems of systems. Journal of Web Engineering, 16(3-4), 212-227.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). The agile manifesto.

Behroozi, N., and Kamandi, A. (2016). Waste elimination of agile methodologies in web engineering. Paper presented at the Web Research (ICWR), 2016 Second International Conference on, 102-107.

Bernaschina, C. (2017). ALMOsT. js: an agile model to model and model to text transformation framework. Paper presented at the International Conference on Web Engineering, 79-97.

Bernaschina, C., Brambilla, M., Koka, T., Mauri, A., and Umuhoza, E. (2017a). Integrating modeling languages and web logs for enhanced user behavior analytics. Paper presented at the Proceedings of the 26th International Conference on World Wide Web Companion, 171-175.

Bernaschina, C., Comai, S., and Fraternali, P. (2017b). IFMLEdit. org: model driven rapid prototyping of mobile apps. Paper presented at the Proceedings of the 4th International Conference on Mobile Software Engineering and Systems, 207-208.

Bernaschina, C., Comai, S., and Fraternali, P. (2017c). Online model editing, simulation and code generation for web and mobile applications. Paper presented at the Proceedings of the 9th International Workshop on Modelling in Software Engineering, 33-39.

Bernaschina, C., Comai, S., and Fraternali, P. (2018). Formal semantics of OMG's Interaction Flow Modeling Language (IFML) for mobile and rich-client application model driven development. Journal of Systems and Software, 137, 239-260.

Blanckaert, J. (2015). Integrating the Interaction Flow Modelling Language (IFML) into the Web Semantics Design Method (WSDM).

Brambilla, M., Cabot, J., and Wimmer, M. (2012). Model-driven software engineering in practice. Synthesis Lectures on Software Engineering, 1(1), 1-182.

Brambilla, M., and Ceri, S. (2018). Modeling, Modeling, Modeling: From Web to Enterprise to Crowd to Social. In A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years (pp. 235-251): Springer.

Brambilla, M., Mauri, A., and Umuhoza, E. (2014). Extending the interaction flow modeling language (ifml) for model driven development of mobile applications front end. Paper presented at the International Conference on Mobile Web and Information Systems, 176-191.

Casteleyn, S., Daniel, F., Dolog, P., and Matera, M. (2009). Engineering web applications (Vol. 30): Springer.

Cetinkaya, D., and Verbraeck, A. (2011). Metamodeling and model transformations in modeling and simulation. Paper presented at the Proceedings of the Winter Simulation Conference, 3048-3058.

Chung, W. (2006). Designing Web-based Business Intelligence Systems: A Framework and Case Studies: In DESRIST. Paper presented at the DESRIST 2006.

De Koch, N. P. (2001). Software engineering for adaptive hypermedia systems. Ph. DThesis, Verlag Uni-Druck, Munich.

Escalona, M. J., and Aragón, G. (2008). NDT. A model-driven approach for web requirements. IEEE Transactions on software engineering, 34(3), 377-390.

Forsberg, K., and Mooz, H. (1991). The relationship of system engineering to the project cycle. Paper presented at the INCOSE International Symposium, 57-65.

Frajták, K., Bures, M., and Jelínek, I. (2015). Using the Interaction Flow Modelling Language for Generation of Automated Front-End Tests. Paper presented at the FedCSIS Position Papers, 117-122.

Ge, X., Paige, R. F., Polack, F. A., Chivers, H., and Brooke, P. J. (2006). Agile development of secure web applications. Paper presented at the Proceedings of the 6th international conference on Web engineering, 305-312.

Grigera, J., Rivero, J. M., Luna, E. R., Giacosa, F., and Rossi, G. (2012). From requirements to web applications in an agile model-driven approach. Paper presented at the International Conference on Web Engineering, 200-214.

Gross, J. M., and McInnis, K. R. (2003). Kanban made simple: demystifying and applying Toyota's legendary manufacturing process: Amacom.

Group, T. O. M. OMG. from https://www.omg.org/

Hamdani, M., Butt, W. H., Anwar, M. W., and Azam, F. (2018). A Systematic Literature Review on Interaction Flow Modeling Language (IFML). Paper presented at the Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences, 134-138.

Hofmann, C., Lauber, S., Haefner, B., and Lanza, G. (2018). Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework. Procedia Manufacturing, 23, 45-50.

Kienle, H. M., and Distante, D. (2014). Evolution of web systems. In Evolving Software Systems (pp. 201-228): Springer.

Kniberg, H., and Skarin, M. (2010). Kanban and Scrum-making the most of both: Lulu. com.

Koren, I., and Klamma, R. (2018). Generation of Web Frontends from API Documentation with Direwolf Interaction Flow Designer. Paper presented at the International Conference on Web Engineering, 492-495.

Kraus, A., Knapp, A., and Koch, N. (2007). Model-Driven Generation of Web Applications in UWE. MDWE, 261.

Laaz, N., and Mbarki, S. (2016a). Combining Ontologies and IFML Models Regarding the GUIs of Rich Internet Applications. Paper presented at the International Conference on Artificial Intelligence: Methodology, Systems, and Applications, 226-236.

Laaz, N., and Mbarki, S. (2016b). Integrating IFML models and owl ontologies to derive UIs web-Apps. Paper presented at the Information Technology for Organizations Development (IT4OD), 2016 International Conference on, 1-6.

Leau, Y. B., Loo, W. K., Tham, W. Y., and Tan, S. F. (2012). Software development life cycle AGILE vs traditional approaches. Paper presented at the International Conference on Information and Network Technology, 162-167.

Luna, E. R., Sánchez-Begínes, J. M., Rivero, J. M., Morales-Trujillo, L., Enríquez, J. G., and Rossi, G. (2018). Challenges for the Adoption of Model-Driven Web Engineering Approaches in Industry. J. Web Eng., 17(3&4), 183-205.

McDonald, A., and Welland, R. (2001a). Agile web engineering (AWE) process. Department of Computing Science Technical Report TR-2001-98, University of Glasgow, Scotland, 2.

McDonald, A., and Welland, R. (2001b). A survey of web engineering in practice. Department of Computing Science Technical Report R-2001-79, University of Glasgow, Scotland, 1.

McDonald, A., and Welland, R. (2001c). Web engineering in practice. Paper presented at the Proceedings of the fourth WWW10 Workshop on Web Engineering, 21-30.

Mellor, S. J., Scott, K., Uhl, A., and Weise, D. (2004). MDA distilled: principles of model-driven architecture: Addison-Wesley Professional.

Mnkandla, E., and Dwolatzky, B. (2004). A survey of agile methodologies. The transactions of the SA institute of electrical engineers, 3, 236-247.

Moreno, N., Romero, J. R., and Vallecillo, A. (2008). An overview of model-driven web engineering and the MDA. In Web Engineering: Modelling and Implementing Web Applications (pp. 353-382): Springer.

Murugesan, S., Deshpande, Y., Hansen, S., and Ginige, A. (2001). Web engineering: A new discipline for development of web-based systems. In Web Engineering (pp. 3-13): Springer.

Nayak, T., and Reddy, B. L. (2013). A Mining Approach for Web Engineering In Respect Of Business Intelligence Application. Proceedings of International Journal of Modern Engineering Research, 3(2), 721-724.

Naziha Laaz, K. W., Samir Mbarki and Dayang N.A. Jawawi. (2018). Comparative Analysis of Interaction Flow Modeling Language Tools. Journal of Computer Science, 14(9).

Nieto, V., Castro, V., Lopez, F., Ferro, R., and Gonzalez, C. (2016). Model Driven Architecture Software and Interaction Flow Modelling Language for Tourism Data Acquisition in Colombia. Paper presented at the Workshop on Engineering Applications, 368-379.

Olsina, L., and Rossi, G. (2002). Measuring Web application quality with WebQEM. Ieee Multimedia, 9(4), 20-29.

OMG: MDA. from https://www.omg.org/mda/

Oxford English Dictionary. from http://www.oed.com/

Patil, P. (2017). Application for Data Mining and Web Data Mining Challenges.

Powell, T. A., Jones, D. L., and Cutts, D. C. (1998). Web site engineering: beyond Web page design: Prentice-Hall, Inc.

Preciado, J. C., Linaje, M., Morales-Chaparro, R., Sanchez-Figueroa, F., Zhang, G., Kroiß, C., et al. (2008). Designing rich internet applications combining uwe and rux-method. Paper presented at the Web Engineering, 2008. ICWE'08. Eighth International Conference on, 148-154.

Punter, T., Kusters, R., Trienekens, J., Bemelmans, T., and Brombacher, A. (2004). The w-process for software product evaluation: A method for goal-oriented implementation of the iso 14598 standard. Software Quality Journal, 12(2), 137-158.

Rivero, J. M., Grigera, J., Rossi, G., Luna, E. R., and Koch, N. (2011). Towards agile model-driven web engineering. Paper presented at the Forum at the Conference on Advanced Information Systems Engineering (CAiSE), 142-155.

Rivero, J. M., Grigera, J., Rossi, G., Luna, E. R., Montero, F., & Gaedke, M. (2014). Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering. Information and Software Technology, 670-687.

Rossi, G., Urbieta, M., Distante, D., Rivero, J. M., and Firmenich, S. (2016). 25 Years of Model-Driven Web Engineering. What we achieved, What is missing. CLEI Electronic Journal, 19(3), 5-57.

Salini, A., Malavolta, I., and Rossi, F. (2016). Leveraging web analytics for automatically generating mobile navigation models. Paper presented at the 2016 IEEE International Conference on Mobile Services (MS), 103-110.

Schmidt, D. C. (2006). Model-driven engineering. COMPUTER-IEEE COMPUTER SOCIETY-, 39(2), 25.

Schön, E.-M., Thomaschewski, J., and Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. Computer Standards & Interfaces, 49, 79-91.

Segue Technologies. from http://www.seguetech.com

Sharma, S., Sarkar, D., and Gupta, D. (2012). Agile processes and methodologies: A conceptual study. International journal on computer science and Engineering, 4(5), 892.

Singh, R., and Bakshi, A. (2013). Need of Agile Development. International Journal of Recent Technology and Engineering, 2(1), 59-61.

Stuurman, S., van Gastel, B. E., and Passier, H. J. (2014). The design of mobile apps: what and how to teach? Paper presented at the Proceedings of the Computer Science Education Research Conference, 93-100.

Sunil Kumar, T., and Suvarchala, K. (2012). A study web data mining challenges and application for information extraction. IOSR Journal of Computer Engineering (IOSRJCE), 7(3), 24-29.

Tanner, M., and Dauane, M. (2017). THE USE OF KANBAN TO ALLEVIATE COLLABORATION AND COMMUNICATION CHALLENGES OF GLOBAL SOFTWARE DEVELOPMENT. Issues in Informing Science & Information Technology, 14.

Torrecilla-Salinas, C., Sedeño, J., Escalona, M., and Mejías, M. (2016). Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review. Information and Software Technology, 71, 92-107.

UWE.

Van Wel, L., and Royakkers, L. (2004). Ethical issues in web data mining. Ethics and Information Technology, 6(2), 129-140.

Vidya, S., and Banumathy, K. (2015). Web Mining-Concepts and Application. International Journal of Computer Science and Information Technologies, 6(4), 3266-3268.

Wakil, K., and Jawawi, D. (2017a). Analyzing interaction flow modeling language in web development lifecycle. International Journal of Advanced Computer Science and Applications, 8(1), 286-293.

Wakil, K., and Jawawi, D. N. (2017b). Comparison between Web Engineering Methods to Develop Multi Web Applications. Journal of Software, 12(10), 783-794.

Wakil, K., and Jawawi, D. N. (2018). A NEW FRAMEWORK FOR USABILITY EVALUATION WEB ENGINEERING METHODS. Journal of Theoretical & Applied Information Technology, 96(2).

WebML.

White, B. (1996). Web document engineering: Stanford Linear Accelerator Centero. Document Number)

Yu, H., Chang, K. C.-C., and Han, J. (2002). Heterogeneous learner for Web page classification. Paper presented at the null, 538.